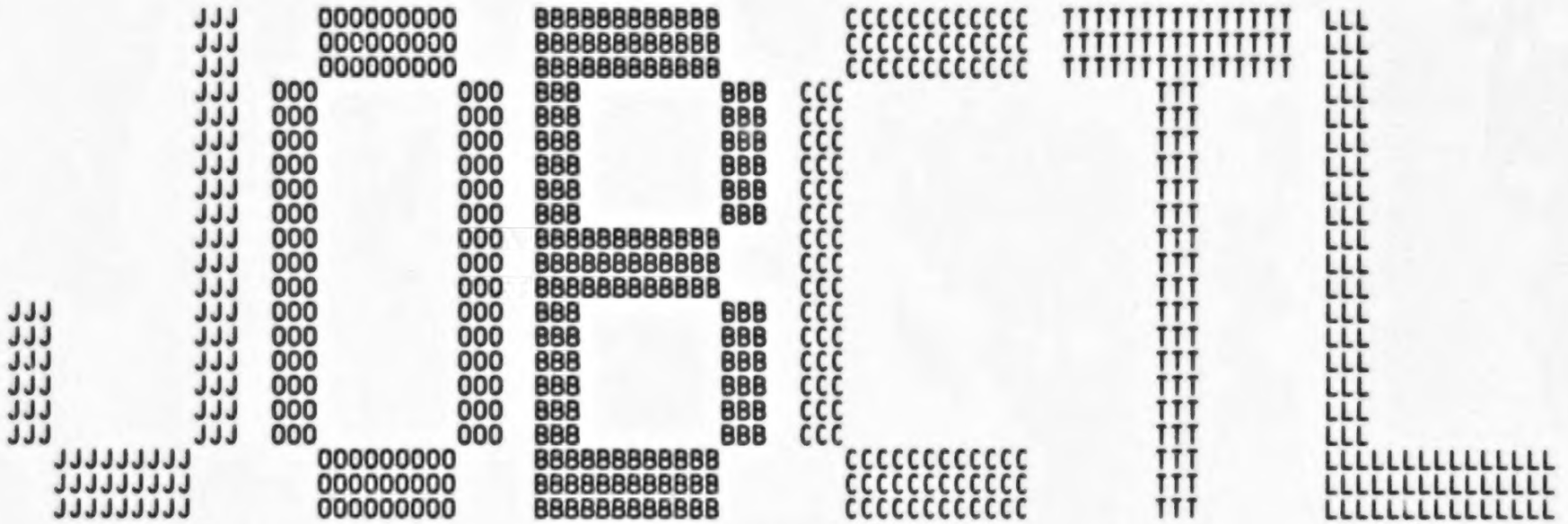
JJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTTTTTTTTTTTTTT LLL
JJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTTTTTTTTTTTTTT LLL
JJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTTTTTTTTTTTTTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBBBBBBBBBBB CCC TTT LLL
JJJ OOO OOO BBBBBBBBBBBB CCC TTT LLL
JJJ OOO OOO BBBBBBBBBBBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJJJJJJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTT LLLLLLLLLLLLLLL
JJJJJJJJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTT LLLLLLLLLLLLLLL
JJJJJJJJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTT LLLLLLLLLLLLLLL

```
          JJ    000000    BBBBBBBB      CCCCCCCC  TTTTTTTTTT  LL          DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
          JJ    000000    BBBBBBBB      CCCCCCCC  TTTTTTTTTT  LL          DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
          JJ  OO      OO  BB      BB  CC            TT      LL          DD      DD  EE          FF
          JJ  OO      OO  BB      BB  CC            TT      LL          DD      DD  EE          FF
          JJ  OO      OO  BB      BB  CC            TT      LL          DD      DD  EE          FF
          JJ  OO      OO  BB      BB  CC            TT      LL          DD      DD  EE          FF
          JJ  OO      OO  BBBBBBBB    CC            TT      LL          DD      DD  EEEEEEEE    FFFFFFFF
          JJ  OO      OO  BBBBBBBB    CC            TT      LL          DD      DD  EEEEEEEE    FFFFFFFF
JJ        JJ  OO      OO  BB      BB  CC            TT      LL          DD      DD  EE          FF
JJ        JJ  OO      OO  BB      BB  CC            TT      LL          DD      DD  EE          FF
JJ        JJ  OO      OO  BB      BB  CC            TT      LL          DD      DD  EE          FF          ....
JJ        JJ  OO      OO  BB      BB  CC            TT      LL          DD      DD  EE          FF          ....
  JJJJJJ      000000    BBBBBBBB      CCCCCCCC      TT      LLLLLLLLLL  DDDDDDDD    EEEEEEEEEE  FF          ....
  JJJJJJ      000000    BBBBBBBB      CCCCCCCC      TT      LLLLLLLLLL  DDDDDDDD    EEEEEEEEEE  FF          ....
```

```
RRRRRRRR    EEEEEEEEEE    QQQQQQ
RRRRRRRR    EEEEEEEEEE    QQQQQQ
RR      RR  EE          QQ      QQ
RR      RR  EE          QQ      QQ
RR      RR  EE          QQ      QQ
RR      RR  EE          QQ      QQ
RRRRRRRR    EEEEEEEE    QQ      QQ
RRRRRRRR    EEEEEEEE    QQ      QQ
RR  RR      EE          QQ  QQ  QQ
RR  RR      EE          QQ  QQ  QQ
RR    RR    EE          QQ    QQ
RR    RR    EE          QQ    QQ
RR      RR  EEEEEEEEEE    QQQQ  QQ
RR      RR  EEEEEEEEEE    QQQQ  QQ
```

```
!       JOBCTLDEF.REQ - Job Controller Common Definitions
!
! Version:      'V04-001'
!
!****************************************************************************
!*                                                                          *
!*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
!*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
!*  ALL RIGHTS RESERVED.                                                    *
!*                                                                          *
!*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
!*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE          *
!*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER     *
!*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
!*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY       *
!*  TRANSFERRED.                                                            *
!*                                                                          *
!*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE    *
!*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT        *
!*  CORPORATION.                                                            *
!*                                                                          *
!*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS     *
!*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
!*                                                                          *
!*                                                                          *
!****************************************************************************
!
!++
! FACILITY:
!       Job controller.
!
! ABSTRACT:
!       This file contains the common definitions for the job controller.
!
! ENVIRONMENT:
!       VAX/VMS user mode.
!--
!
! AUTHOR: M. Jack, CREATION DATE: 21-Feb-1983
!
! MODIFIED BY:
!
!       V04-001 JAK0233         J A Krycka      10-Sep-1984
!               Define FLAGS_V_INVALID_SJH.
!
!       V03-020 JAK0230         J A Krycka      28-Aug-1984
!               Add additonal FLAGS options for debugging purposes.
!
!       V03-019 JAK0228         J A Krycka      20-AUG-1984
!               Define JBC$K_MAXPARSIZ and JBC$K_MAXPARSIZ_ALL.
!
!       V03-018 JAK0224         J A Krycka      10-Aug-1984
!               Define FLAGS_V_NO_REMOTE_DOORBELL.
```

```
        V03-017 JAK0219         J A Krycka      15-Jul-1984
                Update own storage allocation and literal definitions and add
                more diagnostic counters.

        V03-016 JAK0218         J A Krycka      12-Jul-1984
                Remove REQUIRE statement referring to JBCMSG as JBC$ message
                symbols have been moved to STARLET.L32.

        V03-015 JAK0214         J A Krycka      25-May-1984
                Add BUG_CHECK macro.

        V03-014 JAK0210         J A Krycka      10-May-1984
                Add FLAGS_V_QUEUE_CREATED and FLAGS_V_QUEUE_SHARED.

        V03-013 JAK0207         J A Krycka      07-May-1984
                Increase value of JBC$K_QUEUE_MBF and add diagnostic trace
                vectors and diagnostic counters.

        V03-012 JAK0206         J A Krycka      06-May-1984
                Add IMAGE_DUMP_STSFLG.

        V03-011 JAK0203         J A Krycka      17-Apr-1984
                Update MAX_SNDJBC_ITEM value.

        V03-010 JAK0202         J A Krycka      16-Apr-1984
                Add/modify queue file creation parameters.

        V03-009 MLJ0118         Martin L. Jack, 23-Aug-1983
                Change field names, update for new $SJCDEF and $QUIDEF.

        V03-008 MLJ0115         Martin L. Jack, 30-Jul-1983
                Changes for job controller baselevel.

        V03-007 MLJ0114         Martin L. Jack, 23-Jun-1983
                Changes for job controller baselevel.

        V03-006 MLJ0113         Martin L. Jack, 26-May-1983
                Changes for job controller baselevel.

        V03-005 MLJ0112         Martin L. Jack, 29-Apr-1983
                Changes for job controller baselevel.

        V03-004 MLJ0109         Martin L. Jack, 14-Apr-1983
                Changes for job controller baselevel.

        V03-003 MLJ0107         Martin L. Jack, 04-Mar-1983
                Delete JBC$_NORMAL (now in JBCMSG again).

        V03-002 MLJ0106         Martin L. Jack, 01-Mar-1983
                Delete definition of PID_W_PIX.

        V03-001 CWH1002         CW Hobbs        01-Mar-1983
                Define SCH$GL_PIXWIDTH cell and change the PID_W_PIX macro
                to use pixwidth to find the width of the pix in the extended
                process id.
```

```
!
!**
```

```
LIBRARY 'SYS$LIBRARY:LIB';
REQUIRE 'LIB$:SYSQUEDEF';


LITERAL
        TRUE=           1,
        FALSE=          0;


STRUCTURE
        BBLOCK[O,P,S,E;N]=
            [N]
            (BBLOCK + O)<P,S,E>;


PSECT
        CODE=           CODE,
        PLIT=           CODE,
        OWN=            DATA(ADDRESSING_MODE(LONG_RELATIVE)),
        GLOBAL=         DATA;


SWITCHES
        ADDRESSING_MODE(
            EXTERNAL=LONG_RELATIVE,
            NONEXTERNAL=WORD_RELATIVE);
```

```
MACRO

        ! Macros to determine if the value of an expression is one of a set of
        ! specified small-integer values.  These macros can be used only if the
        ! following conditions are met:
        !
        !       The value to be tested is in the range 0 through 127.
        !
        !       The values to be tested for are all in the range 0 through 31.
        !
        ! Example:
        !
        !       IF ONEOF_(.X, BMSK_(1,3,5)) ...
        !
        ! The code generated is much more efficient than a series of comparisons
        ! (provided that the parameters of BMSK_ are all compile-time constant).

        XBMSK_[A]=
            %IF (A) GTRU 31 %THEN %WARN('ONEOF won''t work') %FI
            (1 ^ (31 - (A))) %,


        BMSK_[]=
            (0 OR XBMSK_(%REMAINING)) %,


        ONEOF_(A,B)=
            ((%(B) ^ (A)) LSS 0) %;
```

```
LINKAGE
        L_OUTPUT_1=             CALL( ;
                                    REGISTER=11),

        L_OUTPUT_2=             CALL( ;
                                    REGISTER=10, REGISTER=11),

        L_OUTPUT_3=             CALL( ;
                                    REGISTER=9, REGISTER=10,
                                    REGISTER=11),

        L_OUTPUT_4=             CALL( ;
                                    REGISTER=8, REGISTER=9,
                                    REGISTER=10, REGISTER=11);
```

```
LITERAL
        JBC$K_AFTER_IDT=        1,      ! REQIDT for timed jobs
        JBC$K_HOURLY_IDT=       2,      ! REQIDT for hourly restricted logins
        JBC$K_MINUTE_IDT=       3,      ! REQIDT for minute restricted logins
        JBC$K_QUEUE_ALQ=        100,    ! Allocation/extend quantity for queue file
        JBC$K_QUEUE_GBC=        0,      ! Global buffer count for queue file
        JBC$K_QUEUE_MBC=        1,      ! Multiblock count for queue file
        JBC$K_QUEUE_MBF=        50,     ! Multibuffer count for queue file
        JBC$K_MAXACCREC=        1024,   ! Largest accounting record
        JBC$K_MAXBUFMSG=        100,    ! Maximum number of buffered mailbox messages
        JBC$K_MAXFILERR=        2,      ! Allowed consecutive accounting file errors
        JBC$K_MAXGENTGT=        124,    ! Maximum generic targets per queue
        JBC$K_MAXPAGES=         5,      ! Maximum contiguous VM allocation
        JBC$K_MAXPARSIZ=        255,    ! Maximum size for a single batch parameter string
        JBC$K_MAXPARSIZ_ALL=    480,    ! Maximum size for all batch parameter strings combined
        JBC$K_MBXBUFSIZ=        1024,   ! Job controller mailbox buffer size
        JBC$K_SMBMBXSIZ=        1024,   ! Symbiont mailbox buffer size
        JBC$K_SYNC_EFN=         1;      ! EFN for all synchronous services

LITERAL
        MIN_GETQUI_FUNC=        QUI$_CANCEL_OPERATION,
        MAX_GETQUI_FUNC=        QUI$_RESERVED_FUNC_2,
        MIN_GETQUI_ITEM=        QUI$_ACCOUNT_NAME,
        MAX_GETQUI_ITEM=        QUI$_RESERVED_OUTPUT_6,

        MIN_SNDACC_FUNC=        ACC$K_INSMESG,
        MAX_SNDACC_FUNC=        ACC$K_DISASEL,

        MIN_SNDJBC_FUNC=        SJC$_ABORT_JOB,
        MAX_SNDJBC_FUNC=        SJC$_RESERVED_FUNC_2,
        MIN_SNDJBC_ITEM=        SJC$_ACCOUNTING_MESSAGE,
        MAX_SNDJBC_ITEM=        SJC$_RESERVED_OUTPUT_2,

        MIN_SNDSMB_FUNC=        SMR$K_INITIAL,
        MAX_SNDSMB_FUNC=        SMR$K_SYNCJOB;

BIND
        JBC$_CLOSEOUT=          JBC$_FACILITY^16 + SHR$_CLOSEOUT,
        JBC$_NOCMKRNL=          JBC$_FACILITY^16 + SS$_NOCMKRNL,
        JBC$_NOOPER=            JBC$_FACILITY^16 + SS$_NOOPER,
        JBC$_NOSYSNAM=          JBC$_FACILITY^16 + SS$_NOSYSNAM,
        JBC$_OPENIN=            JBC$_FACILITY^16 + SHR$_OPENIN,
        JBC$_OPENOUT=           JBC$_FACILITY^16 + SHR$_OPENOUT,
        JBC$_READERR=           JBC$_FACILITY^16 + SHR$_READERR,
        JBC$_WRITEERR=          JBC$_FACILITY^16 + SHR$_WRITEERR;
```

```
MACRO
        ACM$T_QUEUE_1=          70,0,0,0 %,     ! Queue name
        ACM$S_QUEUE_1=          16 %,
        ACM$W_ENTRY_NUMBER=     86,0,16,0 %,    ! Job entry number
        ACM$T_QUE_OPT=          86,0,0,0 %,     ! Start of options
        ACM$T_QUEUE_2=          86,0,0,0 %,     ! Second queue name
        ACM$S_QUEUE_2=          16 %,
        ACM$T_QJB_OPT=          88,0,0,0 %,     ! Start of options
        ACM$T_JOBNAME=          88,0,0,0 %,     ! Job name
        ACM$T_ENT_OPT=          134,0,0,0 %,    ! Start of ENTER options
        ACM$T_ADF_OPT=          118,0,0,0 %,    ! Start of ADDFIL options

        ACM$T_INT_FID=          -36,0,0,0 %,    ! Internal FID item
        ACM$W_INT_RSL=          -38,0,16,0 %,   ! Internal file spec length

        ACM$B_RMOD=             37,0,8,0 %,     ! Requestor's access mode
        ACM$L_IMAGECNT=         68,0,32,0 %,    ! Image counter
        ACM$L_EFN=              72,0,32,0 %,    ! Event flag number
        ACM$L_IOSB=             76,0,32,0 %,    ! IOSB address
        ACM$L_ASTADR=           80,0,32,0 %,    ! AST routine address
        ACM$L_ASTPRM=           84,0,32,0 %,    ! AST routine parameter
        ACM$W_FUNC=             88,0,16,0 %,    ! Function code
        ACM$T_ITMLST=           90,0,0,0 %;     ! Start of item list
```

```
! Miscellany.
!
MACRO
        MSG_W_TYPE=             0,0,16,0 %,     ! Mailbox message type
        UIC_W_MEM=              0,0,16,0 %,     ! UIC member number
        UIC_W_GRP=              2,0,16,0 %;     ! UIC group number


! Output from ENQUEUE_JOB, input to JOB_STATUS_MESSAGE.
!
LITERAL
        ENQ_K_CURRENT=          0,              ! Job is in current queue
        ENQ_K_HOLD=             1,              ! Job is in hold queue
        ENQ_K_PENDING=          2,              ! Job is in pending queue
        ENQ_K_TIMER=            3,              ! Job is in timer queue
        ENQ_K_COMPLETE=         4;              ! Job is completing


! General flags.
!
MACRO
        FLAGS_V_QUEUE_LOCKED=   0,0,1,0 %,      ! Queue file is locked
        FLAGS_V_READ_POSTED=    0,1,1,0 %,      ! Mailbox read outstanding
        FLAGS_V_QUEUE_CREATED=  0,2,1,0 %,      ! Queue file was created
        FLAGS_V_QUEUE_SHARED=   0,3,1,0 %,      ! Queue file can be shared
        FLAGS_V_NO_REMOTE_DOORBELL= 0,4,1,0%,   ! No remote doorbell lock defined
        FLAGS_V_INVALID_SJH=    0,5,1,0 %,      ! Job record has been released
        FLAGS_V_FLAG6=          0,6,1,0 %,      ! Spare flag 6
        FLAGS_V_FLAG7=          0,7,1,0 %,      ! Spare flag 7
        FLAGS_V_FLAG8=          0,8,1,0 %,      ! Spare flag 8
        FLAGS_V_FLAG9=          0,9,1,0 %,      ! Spare flag 9
        FLAGS_V_FLAG10=         0,10,1,0 %,     ! Spare flag 10
        FLAGS_V_FLAG11=         0,11,1,0 %;     ! Spare flag 11

! Debugging flags.
!
MACRO
        FLAGS_V_READ_VMSD2=     0,16,1,0%,      ! Reset debugging control flags from VMSD2 sysgen parameter
        FLAGS_V_CLUSTER_SCRAM=  0,17,1,0 %,     ! On fatal error synchronously bug check entire cluster
        FLAGS_V_LEAVE_OPEN=     0,18,1,0 %,     ! On fatal error leave system job queue file open
        FLAGS_V_BUGCHECK=       0,19,1,0 %,     ! On fatal error bug check system vs abort/restart image
        FLAGS_V_CS_QF_DEADLOCK= 0,20,1,0 %,     ! Bug check cluster on queue file deadlock error
        FLAGS_V_LOG_QF_REPAIR=  0,21,1,0 %,     ! Log on-the-fly repair of queue file
        FLAGS_V_OMIT_QF_INIT=   0,22,1,0 %,     ! Omit queue file initialization after opening it
        FLAGS_V_FLAG23=         0,23,1,0 %,     ! Spare flag 23
        FLAGS_V_FLAG24=         0,24,1,0 %,     ! Spare flag 24
        FLAGS_V_FLAG25=         0,25,1,0 %,     ! Spare flag 25
        FLAGS_V_FLAG26=         0,26,1,0 %,     ! Spare flag 26
        FLAGS_V_FLAG27=         0,27,1,0 %;     ! Spare flag 27


! Diagnostic flags.
```

```
!
MACRO
        DIAG_V_FLAG0=           0,0,1,0 %,      ! Diagnostic flag 0
        DIAG_V_FLAG1=           0,1,1,0 %,      ! Diagnostic flag 1
        DIAG_V_FLAG2=           0,2,1,0 %,      ! Diagnostic flag 2
        DIAG_V_FLAG3=           0,3,1,0 %;      ! Diagnostic flag 3


! Fixed/variable data field.
!
MACRO
        FVDF_LENGTH=            0,0,16,0 %,     ! Length of stored data
        FVDF_LINK=              2,0,32,0 %,     ! Link to auxiliary data record
        FVDF_DATA=              2,0,0,0 %;      ! Start of in-place stored data


! Event codes for SCAN_INCOMPLETE_SERVICES.
!
LITERAL
        ISRV_K_REMOTE=          0,              ! Remote node signalled
        ISRV_K_SYNCHRONIZE=     1,              ! Job with SYNCHRONIZE completed
        ISRV_K_SYMBIONT=        2,              ! Symbiont reported completion
        ISRV_K_PURGE_SYSID=     3,              ! Purge references to SYSID
        ISRV_K_PURGE_SMQ=       4,              ! Purge references to SMQ
        ISRV_K_PURGE_SJH=       5;              ! Purge references to SJH


! Flag codes for pause, resume, abort.
!
MACRO
        ISRV_V_ALIGNMENT_MASK=  0,0,1,0 %,      ! Mask alignment data
        ISRV_V_TOP_OF_FILE=     0,1,1,0 %;      ! Position to top of file

LITERAL
        ISRV_M_ALIGNMENT_MASK=  $FIELDMASK(ISRV_V_ALIGNMENT_MASK),
        ISRV_M_TOP_OF_FILE=     $FIELDMASK(ISRV_V_TOP_OF_FILE);


! Output item descriptor.
!
MACRO
        ODSC_W_LENGTH=          0,0,16,0 %,     ! Buffer length
        ODSC_A_POINTER=         2,0,32,0 %,     ! Buffer address
        ODSC_A_LENPOINTER=      6,0,32,0 %;     ! Return length buffer address

LITERAL
        ODSC_S_ENTRY=           10;             ! Size of output item descriptor
```

```
!
! Process data block.
!
MACRO
        PDB_LINK=               0,0,32,0 %,     ! Link to next record
        PDB_COUNT=              4,0,32,0 %,     ! Count of used entries
        PDB_ENTRIES=            8,0,0,0 %,      ! Base of entries
        PDE_PID=                0,0,32,0 %,     ! Process ID
        PDE_TYPE=               4,0,32,0 %,     ! Process type
        PDE_P1=                 8,0,32,0 %,     ! Two longwords of arbitrary
        PDE_P2=                 12,0,32,0 %;    !     information

LITERAL
        PDE_K_ANY=              0,              ! Match any type on find
        PDE_K_BATCH=            1,              ! Batch process
        PDE_K_SYMBIONT=         2,              ! Symbiont process
        PDE_K_OPEN_JOB=         3,              ! Open job for process
        PDE_S_ENTRY=            16,             ! Length of one entry
        PDB_K_MAX=              (512 - $BYTEOFFSET(PDB_ENTRIES)) / PDE_S_ENTRY;
                                                ! Number of entries per block


!
! Interface to SEARCH_QUEUES routine.
!
MACRO
        QSM_V_OPEN=             0,0,1,0 %,      ! Job in open queue
        QSM_V_TIMER=            0,1,1,0 %,      ! Job in timer queue
        QSM_V_PENDING=          0,2,1,0 %,      ! Job in pending queue
        QSM_V_HOLD=             0,3,1,0 %,      ! Job in hold queue
        QSM_V_CURRENT=          0,4,1,0 %;      ! Job in current queue

LITERAL
        QSM_K_CTXSIZE=          9 * 4,          ! Size of context block
        QSM_K_NO_REMOVE=        0,              ! Never dequeue job
        QSM_K_REMOVE=           1,              ! Always dequeue job
        QSM_K_REMOVE_INACTIVE=  2,              ! Dequeue if not executing
        QSM_M_OPEN=             $FIELDMASK(QSM_V_OPEN),
        QSM_M_TIMER=            $FIELDMASK(QSM_V_TIMER),
        QSM_M_PENDING=          $FIELDMASK(QSM_V_PENDING),
        QSM_M_HOLD=             $FIELDMASK(QSM_V_HOLD),
        QSM_M_CURRENT=          $FIELDMASK(QSM_V_CURRENT);


!
! Response message returned to mailbox from $SNDSMB and $SNDACC.
!
MACRO
        RSP_W_TYPE=             0,0,16,0 %,     ! Mailbox message type
        RSP_W_ENTRY_NUMBER=     2,0,16,0 %,     ! Entry number
        RSP_L_STATUS=           4,0,32,0 %;     ! Completion status

LITERAL
        RSP_S_MESSAGE=          8;              ! Length of message
```

```
! Short descriptors.
!
MACRO
        SDSC_W_LENGTH=          0,0,16,0 %,     ! Length of string
        SDSC_A_POINTER=         2,0,32,0 %;     ! Address of string

LITERAL
        SDSC_K_LENGTH=          6;              ! Block length
```

```
MACRO
        CLEAR_SYSID(S1)=
            BEGIN
            (S1) = 0;
            ((S1)+4)<0,16> = 0;
            END %,

        COPY_SYSID(S1,S2)=
            BEGIN
            (S2) = .(S1);
            ((S2)+4)<0,16> = .((S1)+4)<0,16>;
            END %,

        SYSID_EQL(S1,S2)=
            BEGIN
            .(S1) EQL .(S2) AND .((S1)+4)<0,16> EQL .((S2)+4)<0,16>
            END %,

        SYSID_NEQ(S1,S2)=
            BEGIN
            .(S1) NEQ .(S2) OR .((S1)+4)<0,16> NEQ .((S2)+4)<0,16>
            END %,

        CLEAR_TIME(T1)=
            BEGIN
            (T1) = 0;
            (T1)+4 = 0;
            END %,

        COPY_TIME(T1,T2)=
            BEGIN
            (T2) = .(T1);
            (T2)+4 = .((T1)+4);
            END %,

        TIME_GTRU(T1,T2)=
            BEGIN
            .((T1)+4) GTRU .((T2)+4) OR
            (.((T1)+4) EQL .((T2)+4) AND .(T1) GTRU .(T2))
            END %,

        TIME_GEQU(T1,T2)=
            BEGIN
            .((T1)+4) GTRU .((T2)+4) OR
            (.((T1)+4) EQL .((T2)+4) AND .(T1) GEQU .(T2))
            END %;
```

```
MACRO
        VALUE_DECL_(A)=
            ALIGN(0) A %,

        VALUE_DECL_DESC_=
            ALIGN(0) BBLOCK[SDSC_K_LENGTH] %;


MACRO
        BUG_CHECK(CODE)=
            BEGIN
            BUILTIN BUGW;
            EXTERNAL LITERAL %NAME('BUG$_',CODE);
            BUGW (%NAME('BUG$_',CODE) OR 4)
            END %;


PSECT
        OWN=COMMON(OVERLAY,ADDRESSING_MODE(LONG_RELATIVE));


OWN
        DIAG_STORAGE_BASE:      VECTOR[0],              ! Start of diagnostic area
        DIAG_TRACE:             VECTOR[24],             ! Diagnostic trace values
        DIAG_COUNT:             VECTOR[24],             ! Diagnostic I/O related counters
        DIAG_FLAGS:             BBLOCK[4],              ! Diagnostic flags
        WORK_AREA:              VECTOR[11],             ! Scratch work area
        SNDJBC_COUNT:           VECTOR[MAX_SNDJBC_FUNC+1], ! Number of $SNDJBC function code requests
        GETQUI_COUNT:           VECTOR[MAX_GETQUI_FUNC+1], ! Number of $GETQUI function code requests
        SNDACC_COUNT:           VECTOR[MAX_SNDACC_FUNC+1], ! Number of $SNDACC function code requests
        SNDSMB_COUNT:           VECTOR[MAX_SNDSMB_FUNC+1], ! Number of $SNDSMB function code requests
        DIAG_STORAGE_END:       VECTOR[0],              ! End of diagnostic area

        FLAGS:                  BBLOCK[4],              ! General flags
        IMAGE_DUMP_STSFLG,                              ! Image dump flag for $CREPRC
        THIS_SYSID:             BBLOCK[6],              ! System ID of this system
        CUR_TIME:               VECTOR[2],              ! Current time
        HOURLY_TIME:            VECTOR[2],              ! Time of next hourly timer expiration
        HOURLY_PARAMS:          VECTOR[5],              ! Parameters for hourly $CMKRNL routine
        SYMBIONT_COUNT,                                 ! Number to append to symbiont process name
        QUEUE_REFERENCE_COUNT,                          ! Number of reasons queue file must stay open
        MBX_MESSAGE_COUNT,                              ! Number of buffered mailbox messages
        MBX:                    REF BBLOCK,             ! Pointer to current mailbox message
        MBX_END:                REF BBLOCK,             ! Pointer past end of current mailbox message
        MEMORY_FREE_QUEUES:     VECTOR[2*JBC$K_MAXPAGES], ! Queue headers for memory free lists
        NONAST_WORK_QUEUE:      VECTOR[2],              ! Queue header for non-AST work queue
        BCB_FREE_LIST,                                  ! List of free buffer control blocks
        BCB_ACTIVE_LIST,                                ! List of active buffer control blocks
        GQL_FREE_LIST,                                  ! List of free $GETQUI lock blocks
        GQL_ACTIVE_LIST,                                ! List of active $GETQUI lock blocks
        OPEN_GETQUI_LIST,                               ! List of open $GETQUI operation blocks
        PROCESS_DATA_LIST,                              ! List of process data blocks
        SYMBIONT_CONTROL,                               ! List of symbiont control blocks
        SPARE_AREA:             VECTOR[3],              ! Spare work area
        REMOTE_REQUEST_LKSB:    VECTOR[4,WORD],         ! Lock status block for remote request lock
        QUEUE_FILE_LKSB:        VECTOR[4,WORD],         ! Lock status block for queue master lock
```

```
        QUEUE_LOCK_LKSB:        VECTOR[4,WORD],         ! Lock status block for queue synchronization lock
        RSP:                    BBLOCK[RSP_S_MESSAGE],  ! Service response ($SNDACC/$SNDSMB format)
        JBC_PRIORITY,                                   ! Job controller's base priority
        JBC_PRIVILEGES:         BBLOCK[8],              ! Job controller's privileges
        JBC_QUOTAS:             BBLOCK[66],             ! Job controller's quotas
        JBC_UIC,                                        ! Job controller's UIC

        QUEUE_FAB:              $FAB_DECL,              ! FAB for queue file
        QUEUE_RAB:              $RAB_DECL,              ! RAB for queue file
        QUEUE_NAM:              $NAM_DECL,              ! NAM block for queue file
        QUEUE_XAB:              $XABPRO_DECL,           ! Protection XAB for queue file
        QUEUE_RSA:              VECTOR[NAM$C_MAXRSS,BYTE], ! Resultant string for queue file
        QUEUE_ALQ:              BBLOCK[4],              ! Queue file allocation and extend quantity
        QUEUE_MBF:              BBLOCK[1],              ! Queue file multibuffer count

        ACCOUNTING_FABS:        VECTOR[2],              ! Pointers to current, inactive accounting FABs
        ACCOUNTING_RABS:        VECTOR[2],              ! Pointers to current, inactive accounting RABs
        ACCOUNT_FAB_A:          $FAB_DECL,              ! FAB for accounting file (primary)
        ACCOUNT_RAB_A:          $RAB_DECL,              ! RAB for accounting file (primary)
        ACCOUNT_NAM_A:          $NAM_DECL,              ! NAM block for accounting file (primary)
        ACCOUNT_RSA_A:          VECTOR[NAM$C_MAXRSS,BYTE], ! Resultant string for accounting file (primary)
        ACCOUNT_FAB_B:          $FAB_DECL,              ! FAB for accounting file (alternate)
        ACCOUNT_RAB_B:          $RAB_DECL,              ! RAB for accounting file (alternate)
        ACCOUNT_NAM_B:          $NAM_DECL,              ! NAM block for accounting file (alternate)
        ACCOUNT_RSA_B:          VECTOR[NAM$C_MAXRSS,BYTE], ! Resultant string for accounting file (alternate)

        DIAG_FAB:               $FAB_DECL,              ! FAB for diagnostic file
        DIAG_RAB:               $RAB_DECL,              ! RAB for diagnostic file

        MBX_CHAN,                                       ! Channel assigned to job controller mailbox
        MBX_IOSB:               VECTOR[4,WORD],         ! I/O status block for mailbox read
        MBX_BUFFER:             BBLOCK[JBC$K_MBXBUFSIZ], ! Mailbox read buffer -- MUST follow MBX_IOSB!!!

        VALUE_STORAGE_BASE:             VALUE_DECL_(VECTOR[0]),
        ITEM_PRESENT:                   VALUE_DECL_(BITVECTOR[256]),
        VALUE_GETQUI_BASE:              VALUE_DECL_(VECTOR[0]),
        VALUE_ACCOUNTING_MESSAGE:       VALUE_DECL_DESC,
        VALUE_ACCOUNTING_TYPES:         VALUE_DECL_(BBLOCK[4]),
        VALUE_AFTER_TIME:               VALUE_DECL_(VECTOR[2]),
        VALUE_ALIGNMENT_PAGES:          VALUE_DECL_(BYTE),
        VALUE_BASE_PRIORITY:            VALUE_DECL_(BYTE),
        VALUE_BATCH_INPUT:              VALUE_DECL_DESC,
        VALUE_BATCH_OUTPUT:             VALUE_DECL_(BBLOCK[ODSC_S_ENTRY]),
        VALUE_BUFFER_COUNT:             VALUE_DECL_(BYTE),
        VALUE_CHARACTERISTIC_NAME:      VALUE_DECL_DESC,
        VALUE_CHARACTERISTIC_NUMBER:    VALUE_DECL_(BYTE),
        VALUE_CHARACTERISTICS:          VALUE_DECL_(BITVECTOR[128]),
        VALUE_CHECKPOINT_DATA:          VALUE_DECL_DESC,
        VALUE_CLI:                      VALUE_DECL_DESC,
        VALUE_CPU_DEFAULT:              VALUE_DECL_(LONG),
        VALUE_CPU_LIMIT:                VALUE_DECL_(LONG),
        VALUE_DESTINATION_QUEUE:        VALUE_DECL_(VECTOR[2]),
        VALUE_DEVICE_NAME:              VALUE_DECL_DESC,
        VALUE_ENTRY_NUMBER:             VALUE_DECL_(LONG),
        VALUE_ENTRY_NUMBER_OUTPUT:      VALUE_DECL_(BBLOCK[ODSC_S_ENTRY]),
        VALUE_EXTEND_QUANTITY:          VALUE_DECL_(WORD),
```

```
        VALUE_FILE_COPIES:              VALUE_DECL_(BYTE),
        VALUE_FILE_IDENTIFICATION:      VALUE_DECL_(BBLOCK[36]),
        VALUE_FILE_SETUP_MODULES:       VALUE_DECL_DESC_,
        VALUE_FILE_SPECIFICATION:       VALUE_DECL_DESC_,
        VALUE_FIRST_PAGE:               VALUE_DECL_(LONG),
        VALUE_FORM_DESCRIPTION:         VALUE_DECL_DESC_,
        VALUE_FORM_LENGTH:              VALUE_DECL_(BYTE),
        VALUE_FORM_MARGIN_BOTTOM:       VALUE_DECL_(BYTE),
        VALUE_FORM_MARGIN_LEFT:         VALUE_DECL_(WORD),
        VALUE_FORM_MARGIN_RIGHT:        VALUE_DECL_(WORD),
        VALUE_FORM_MARGIN_TOP:          VALUE_DECL_(BYTE),
        VALUE_FORM_NAME:                VALUE_DECL_DESC_,
        VALUE_FORM_NUMBER:              VALUE_DECL_(LONG),
        VALUE_FORM:                     VALUE_DECL_(VECTOR[2]),
        VALUE_FORM_SETUP_MODULES:       VALUE_DECL_DESC_,
        VALUE_FORM_STOCK:               VALUE_DECL_DESC_,
        VALUE_FORM_WIDTH:               VALUE_DECL_(WORD),
        VALUE_GENERIC_TARGET:           VALUE_DECL_(VECTOR[1+2*JBC$K_MAXGENTGT]),
        VALUE_JOB_COPIES:               VALUE_DECL_(BYTE),
        VALUE_JOB_LIMIT:                VALUE_DECL_(BYTE),
        VALUE_JOB_NAME:                 VALUE_DECL_DESC_,
        VALUE_JOB_RESET_MODULES:        VALUE_DECL_DESC_,
        VALUE_JOB_SIZE_MAXIMUM:         VALUE_DECL_(LONG),
        VALUE_JOB_SIZE_MINIMUM:         VALUE_DECL_(LONG),
        VALUE_JOB_STATUS_OUTPUT:        VALUE_DECL_(BBLOCK[ODSC_S_ENTRY]),
        VALUE_LAST_PAGE:                VALUE_DECL_(LONG),
        VALUE_LIBRARY_SPECIFICATION:    VALUE_DECL_DESC_,
        VALUE_LOG_QUEUE:                VALUE_DECL_(VECTOR[2]),
        VALUE_LOG_SPECIFICATION:        VALUE_DECL_DESC_,
        VALUE_NOTE:                     VALUE_DECL_DESC_,
        VALUE_OPERATOR_REQUEST:         VALUE_DECL_DESC_,
        VALUE_OWNER_UIC:                VALUE_DECL_(LONG),
        VALUE_PAGE_SETUP_MODULES:       VALUE_DECL_DESC_,
        VALUE_PARAMETER_1:              VALUE_DECL_DESC_,
        VALUE_PARAMETER_2:              VALUE_DECL_DESC_,
        VALUE_PARAMETER_3:              VALUE_DECL_DESC_,
        VALUE_PARAMETER_4:              VALUE_DECL_DESC_,
        VALUE_PARAMETER_5:              VALUE_DECL_DESC_,
        VALUE_PARAMETER_6:              VALUE_DECL_DESC_,
        VALUE_PARAMETER_7:              VALUE_DECL_DESC_,
        VALUE_PARAMETER_8:              VALUE_DECL_DESC_,
        VALUE_PRIORITY:                 VALUE_DECL_(BYTE),
        VALUE_PROCESSOR:                VALUE_DECL_DESC_,
        VALUE_PROTECTION:               VALUE_DECL_(LONG),
        VALUE_QUEUE:                    VALUE_DECL_DESC_,
        VALUE_QUEUE_FILE_SPECIFICATION: VALUE_DECL_DESC_,
        VALUE_RELATIVE_PAGE:            VALUE_DECL_(LONG),
        VALUE_RESERVED_INPUT_1:         VALUE_DECL_(BYTE),
        VALUE_RESERVED_INPUT_2:         VALUE_DECL_(WORD),
        VALUE_RESERVED_INPUT_3:         VALUE_DECL_(LONG),
        VALUE_RESERVED_INPUT_4:         VALUE_DECL_DESC_,
        VALUE_RESERVED_OUTPUT_1:        VALUE_DECL_(BBLOCK[ODSC_S_ENTRY]),
        VALUE_RESERVED_OUTPUT_2:        VALUE_DECL_(BBLOCK[ODSC_S_ENTRY]),
        VALUE_SEARCH_STRING:            VALUE_DECL_DESC_,
        VALUE_SCSNODE_NAME:             VALUE_DECL_(BBLOCK[6]),
        VALUE_WSDEFAULT:                VALUE_DECL_(WORD),
```

```
        VALUE_WSEXTENT:                 VALUE_DECL_(WORD),
        VALUE_WSQUOTA:                  VALUE_DECL_(WORD),
        VALUE_STORAGE_END:              VALUE_DECL_(VECTOR[0]);

PSECT
        OWN=DATA;
```

LASWEEP
LIS

XFDEF
FOR

LASNDLDRQ
LIS

LABUFFER
LIS

JOBCTL

JOBCTL
MAP

JOBCTLDEF
REQ

DRSUP
LIS

SYSQUEDEF
SDL

IOSUP

ACCOUNTNG
LIS

DRDEF
MAR